What is claimed is:

1. A computer program product for providing fast and efficient address lookup for an address comprised of a plurality of address components and wherein each address component is deemed to be more significant than its next-sequential neighboring address component, the computer program product embodied on one or more computer-readable media and comprising:

computer-readable program code means for creating a plurality of arrays comprising an array for each of the address components, wherein each array comprises a plurality of entries which are indexed using values of the address component for which the array was created, further comprising: comprising:

computer-readable program code means for obtaining a particular address value to be represented in the plurality of arrays;

computer-readable program code means for obtaining a bit mask associated with the particular address value;

computer-readable program code means for indexing into a highest-order one of the arrays using a most-significant component of the particular address value as an index element;

computer-readable program code means for setting a flag associated with the index element to on if the bit mask indicates that the next-sequential neighboring address component is considered significant, and for setting the flag to off otherwise; and

computer-readable program code means for repeating the indexing and setting while the bit mask indicates that the next-sequential neighboring address component is considered significant, and for (1) storing information associated with the particular address value in a storage or memory location and (2) setting a pointer field associated with the index element to point to the storage or memory location, otherwise; and

computer-readable program code means for retrieving the stored information associated with a selected address value from the plurality of arrays, further comprising:

computer-readable program code means for obtaining the selected address value;

computer-readable program code means for obtaining a selected bit mask associated with the selected address value;

computer-readable program code means for indexing into the highest-order one of the arrays using the most-significant component of the selected address value as the index element; and

computer-readable program code means for determining that no result is available if the index element has no stored information, and for continuing otherwise, wherein the continuing further comprises:

computer-readable program code means for checking the flag associated with the index element; and

computer-readable program code means for returning the stored information from the storage or memory location pointed to by the pointer field when the flag is set off or for repeating the indexing and determining when the flag is set on.

2. The computer program product according to Claim 1, wherein the computer-readable program code means for repeating further comprises computer-readable program code means for setting a use count associated with the storage or memory location to a number which represents a count of the array entries which point to this storage or memory location when the next-

sequential neighboring address component is not considered significant.

3. The computer program product according to Claim 2, wherein the stored information in the memory or storage location comprises an associated bit mask and wherein the computer-readable program code means for retrieving further comprises computer-readable program code means for resolving a collision, further comprising:

computer-readable program code means for comparing the selected address value to each bit mask associated with the stored information from multiple storage or memory locations, yielding a plurality of bit mask results; and

computer-readable program code means for selecting a collision result using that one of the bit mask results which both (1) matches the selected address value according to the selected bit mask and (2) has the longest associated bit mask.

4. The computer program product according to Claim 1, wherein the address is an Internet Protocol (IP) address.

5. The computer program product according to Claim 4, wherein the IP address is an IP version 4 address and wherein there are 4 components in each IP version 4 address and thus 4 arrays.

6. The computer program product according to Claim 4, wherein the IP address is an IP version 6 address and wherein there are 16 address components in each IP version 6 address and

thus 16 arrays.

7. A system for providing fast and efficient address lookup for an address comprised of a plurality of address components and wherein each address component is deemed to be more significant than its next-sequential neighboring address component, the system comprising:

means for creating a plurality of arrays comprising an array for each of the address components, wherein each array comprises a plurality of entries which are indexed using values of the address component for which the array was created, further comprising:

means for obtaining a particular address value to be represented in the plurality of arrays;

means for obtaining a bit mask associated with the particular address value;

means for indexing into a highest-order one of the arrays using a most-significant component of the particular address value as an index element;

means for setting a flag associated with the index element to on if the bit mask indicates that the next-sequential neighboring address component is considered significant, and for setting the flag to off otherwise; and

means for repeating the indexing and setting while the bit mask indicates that the next-sequential neighboring address component is considered significant, and for (1) storing information associated with the particular address value in a storage or memory location and (2) setting a pointer field associated with the index element to point to the storage or memory location, otherwise; and

means for retrieving the stored information associated with a selected address value from the plurality of arrays, further comprising:

means for obtaining the selected address value;

means for obtaining a selected bit mask associated with the selected address value;

means for indexing into the highest-order one of the arrays using the most-significant component of the selected address value as the index element; and

means for determining that no result is available if the index element has no stored information, and for continuing otherwise, wherein the continuing further comprises:

means for checking the flag associated with the index element; and

means for returning the stored information from the storage or memory location pointed to by the pointer field when the flag is set off or for repeating the indexing and determining when the flag is set on.

8. The system according to Claim 7, wherein the means for repeating further comprises means for setting a use count associated with the storage or memory location to a number which represents a count of the array entries which point to this storage or memory location when the next-sequential neighboring address component is not considered significant.

9. The system according to Claim 8, wherein the stored information in the memory or storage location comprises an associated bit mask and wherein the means for retrieving further comprises means for resolving a collision, further comprising:

means for comparing the selected address value to each bit mask associated with the stored information from multiple storage or memory locations, yielding a plurality of bit mask results; and

means for selecting a collision result using that one of the bit mask results which both (1) matches the selected address value according to the selected bit mask and (2) has the longest associated bit mask.

10. The system according to Claim 7, wherein the address is an Internet Protocol (IP) address.

11. The system according to Claim 10, wherein the IP address is an IP version 4 address and wherein there are 4 components in each IP version 4 address and thus 4 arrays.

12. The system according to Claim 10, wherein the IP address is an IP version 6 address and wherein there are 16 address components in each IP version 6 address and thus 16 arrays.

13. A method for providing fast and efficient address lookup for an address comprised of a plurality of address components and wherein each address component is deemed to be more significant than its next-sequential neighboring address component, the method comprising the steps of:

creating a plurality of arrays comprising an array for each of the address components, wherein each array comprises a plurality of entries which are indexed using values of the address component for which the array was created, further comprising the steps of:

obtaining a particular address value to be represented in the plurality of arrays;

obtaining a bit mask associated with the particular address value;

indexing into a highest-order one of the arrays using a most-significant component of the particular address value as an index element;

setting a flag associated with the index element to on if the bit mask indicates that the next-sequential neighboring address component is considered significant, and setting the flag to off otherwise; and

repeating the indexing and setting while the bit mask indicates that the next-sequential neighboring address component is considered significant, and (1) storing information associated with the particular address value in a storage or memory location and (2) setting a pointer field associated with the index element to point to the storage or memory location, otherwise; and

retrieving the stored information associated with a selected address value from the plurality of arrays, further comprising the steps of:

obtaining the selected address value;

obtaining a selected bit mask associated with the selected address value;

indexing into the highest-order one of the arrays using the most-significant component of the selected address value as the index element; and

determining that no result is available if the index element has no stored information, and continuing otherwise, wherein the continuing further comprises the steps of:

checking the flag associated with the index element; and

returning the stored information from the storage or memory location pointed to by the pointer field when the flag is set off or repeating the indexing and determining when the flag is set on.

14. The method according to Claim 13, wherein the repeating step further comprises the step of setting a use count associated with the storage or memory location to a number which represents a count of the array entries which point to this storage or memory location when the next-sequential neighboring address component is not considered significant.

15. The method according to Claim 14, wherein the stored information in the memory or storage location comprises an associated bit mask and wherein the retrieving step further comprises resolving a collision, further comprising the steps of:

comparing the selected address value to each bit mask associated with the stored information from multiple storage or memory locations, yielding a plurality of bit mask results; and

selecting a collision result using that one of the bit mask results which both (1) matches the selected address value according to the selected bit mask and (2) has the longest associated bit mask.

16. The method according to Claim 13, wherein the address is an Internet Protocol (IP) address.

17. The method according to Claim 16, wherein the IP address is an IP version 4 address and wherein there are 4 components in each IP version 4 address and thus 4 arrays.

18. The method according to Claim 16, wherein the IP address is an IP version 6 address and wherein there are 16 address components in each IP version 6 address and thus 16 arrays.

19. A method for providing fast and efficient address lookup for an address comprised of a plurality of address components, the method comprising the steps of:

creating a plurality of arrays comprising an array for each of the address components, wherein each array comprises a plurality of entries which are indexed using values of the address component for which the array was created;

storing entries and information for each address to be subsequently looked up, further comprising the steps of:

creating an entry for a particular address using the plurality of arrays; and

storing information associated with the particular address value in a storage or memory location associated with a last significant component of the entry, wherein the last significant component is determined by a bit mask associated with the particular address; and

retrieving the stored information associated with a selected address value from the plurality of arrays.